.u Visual - Ubiquitous Visual Terminal -

Fujitsu Laboratories Ltd.

What is demonstrated

Characteristics

A triple play function of broadcast receiving, telephony and internet access with high level of quality is realized by the Fujitsu media processor FR-V and Linux OS.

Embedded with multi-function applications

- Full Browser (called Inspirium)
- AV Player
- Wireless IP video telephone
- Wireless IP video transceiver

High speed sending and receiving of moving pictures

IP video phone and wireless IP video transceiver with a capability of sending and receiving QVGA-sized pictures at a speed of 15 frames per second (industry-leading levels of performance).

Specifications

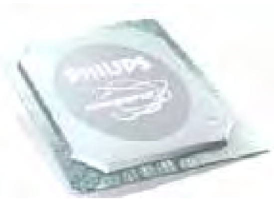
items	Specification
Size/Weight	165x73x26mm/245g (battery included)
Main LSI	CPU: FR461 made by Fujitsu (400MHz, 8 parallel operations per cycle) SDRAM 128MB, Flash ROM 64MB
Display	3.7inch VGA(640x480) TFT Color LCD
Camera	CMOS 0.35 million pixel
Wireless LAN	Wireless LAN module (IEEE802.11b compliant) embedded
External interface	CF、SDIO、USB (in the cradle)
OS	Embedded Linux for FR-V
GUI	X-Window System, WideStudio/MWT
Middleware Application	• MPEG-4 codec • AAC codec • AVC/H.264 dec • Wireless IP video phone • wireless IP video transceiver

Graphics Subsystem in an Embedded World

What is demonstrated

A DirectFB and UHAPI compliant DTV platform playing an ATSC HD stream

System Information

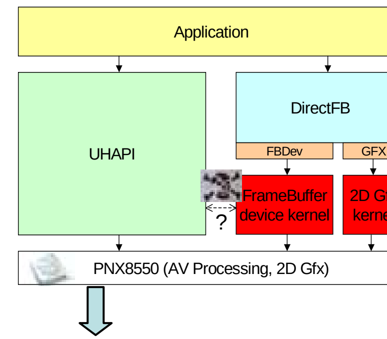


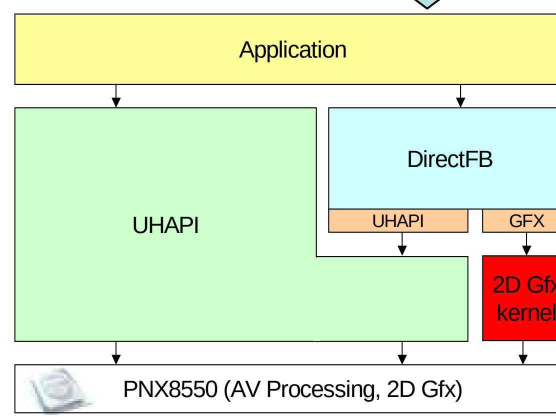
Philips TV810 Hybrid DTV with PNX8550

- MIPS/250MHz runs Linux 2.6.10
- TriMedia media processing cores
- Dedicated media and 2D Gfx hardware accelerators
- Integrated DTV/DMA middleware from AVTREX

How was the Linux improved

By integrating DirectFB and UHAPI directly, instead of using the Linux FB device.





Simplified architecture and implementation

- Avoids going through very limited FB API layer
- Solves resource management issues
- Much improved interoperability
- Modular approach

VideoClip Player and Linux

Armin Gerritsen / Philips Semiconductors

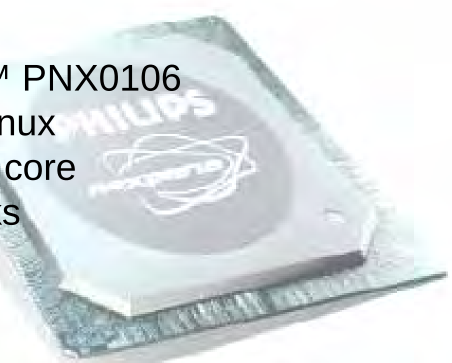
PHILIPS

What is demonstrated

Qtopia-frontend on VCP-1/PNX0106

Hardware Information

Philips VCP-1 with Nexperia™ PNX0106
- o ARM926/133MHz runs Linux
- o EPICS media processing core
- o Various connectivity hooks
- o 32MByte system memory



How was the Linux used

Linux played an essential role:
- o Used during prototyping
- o Used as main OS

Many different use-cases:
- o Playback on TV-out and LCD
- o Many connectivity hooks

Bluetooth USB UPnP WiFi nexperia PHILIPS

MythTV on Nexperia

Klaas de Waal / Philips Semiconductors

PHILIPS

What is demonstrated

MythTV-frontend on PNX8550 playing HD content

Hardware Information

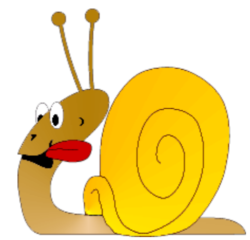
Philips STB810 IP-STB with PNX8550

- o MIPS/250MHz runs Linux
- o TriMedia media processing cores
- o Dedicated hardware accelerators
- o 128MByte system memory

How was the Linux improved

Use native compilation on MIPS:

- o Avoids cross-compilation problems
- o Only configure/make/make install!!



Improve compilation speed with:

- o distcc for distributed compilation
- o Use PC as “compute server”

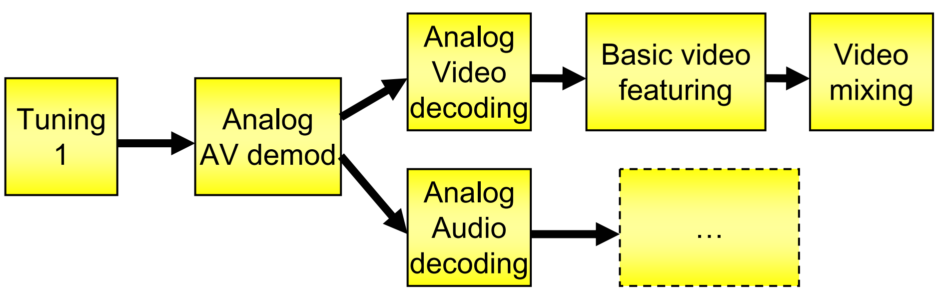
PHILIPS

UHAPI4Linux : open source implementation of UHAPI

Ruud Derwig

What is demonstrated

An open source implementation of UHAPI.
UHAPI is an open standard for the audio/video streaming API.
The implementation supports a number of UHAPI "logical components", among others the ones shown in the streaming graph below.



Hardware Information

Standard PC with standard tuner card.

How was the Linux improved

Up to now Linux lacked a complete, well-defined, and consistent API for audio/video streaming. UHAPI fills this gap, and this open source implementation is an example implementation of UHAPI.
The (proposed) CELF Audio/Video/Graphics 2.0 specification uses UHAPI as the main interface for controlling audio/video and DirectFB/OpenGL ES for graphics.

Availability

http://sourceforge.net/projects/uhapi4linux/
UHAPI specification: www.uhapi.org
(and also on the CELF wiki pages)

Embedded Linux based Terrestrial DMB TV

Samsung Electronics

What is demonstrated

- DMB(Digital Multimedia Broadcasting) Demo Environment
  - System
    - + MPEG2, MPEG4 System
  - Video
    - + coding scheme: MPEG4 Part-10 H.264
    - + resolution: QVGA size (320X240)
    - + frame rate: 30 (f/s)
  - Audio
    - + coding scheme: MPEG4-BSAC
    - + FM stereo quality sound
  - Total bitrate under 500kbps
  - One ensemble of 1.5Mhz in VHF 6Mhz TV channel

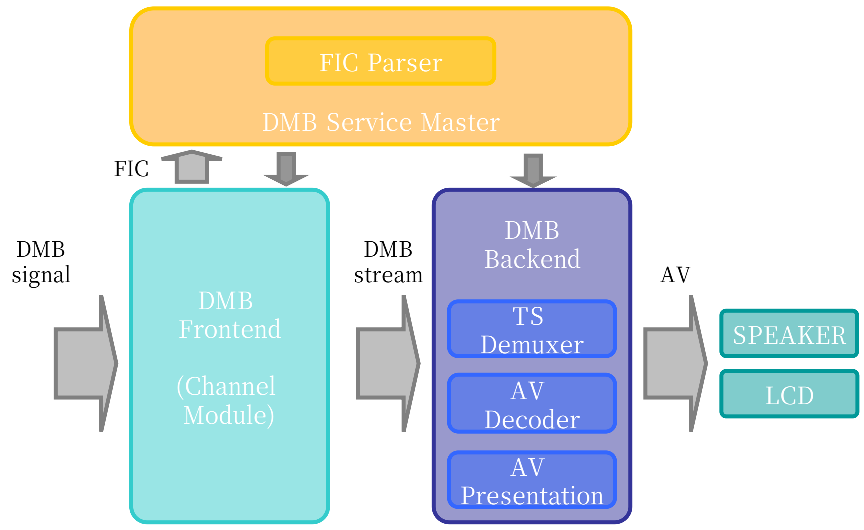
- DMB TV Receiver
  - Linux 2.6.11 kernel
  - Multithread DMB TV application

Hardware Information

- DMB HW : Channel module from Samsung
- CPU : AU1200 (MIPS32 core SOC) from AMD
- RAM : 128MBytes DDR SDRAM
- Flash Memory : 64MBytes onenand

How was the Linux improved

- DMB SW Diagram implemented in Linux



Patch Availability

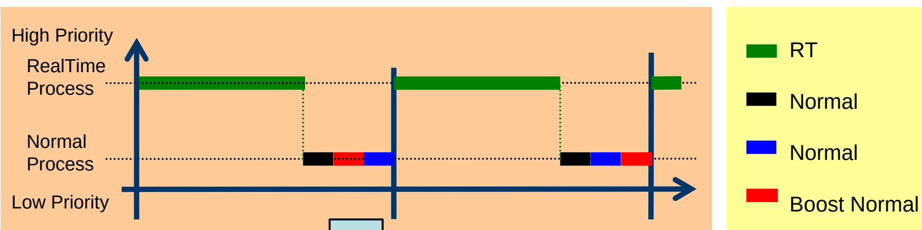
Patch release schedule is not decided yet.

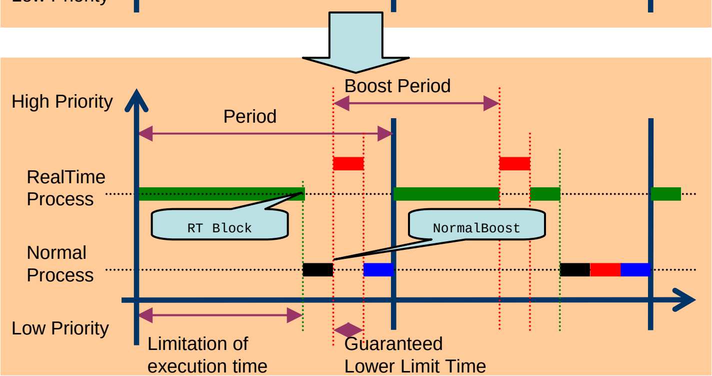
Linux Kernel CPU Resource Reservation

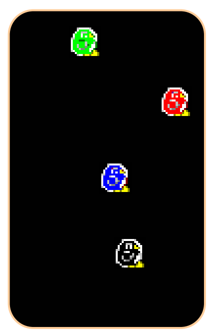
Hitachi, Lineo Solutions

What is demonstrated

Linux assigns higher priority to a real-time process than a normal process so that no other processes could run if some real-time process would not release the CPU resource.
In order to realize comfortable GUI in embedded systems like DTV which consists of real-time processes, interactive processes and background processes, we need to assign the CPU resource to a particular process which takes care of GUI.
We implement CPU Resource Reservation Feature which specifies Upper Limit as well as Lower Limit of CPU usage for a process so that we could get response in acceptable time from a particular process.







Demonstration

Hardware Information

RealView Versatile ARM926EJ-S

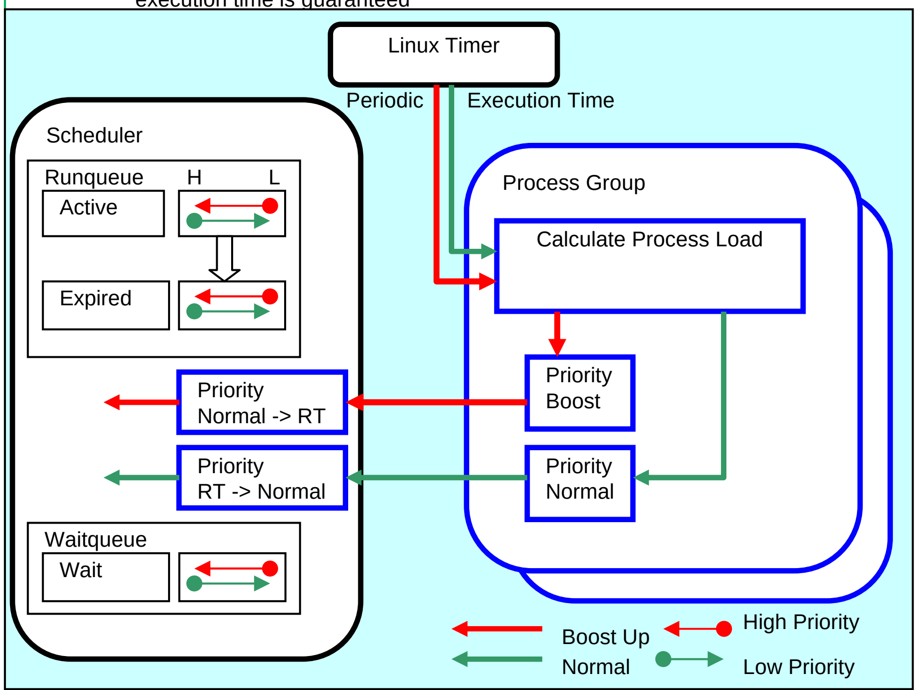
How was the Linux improved

Block of RT

RT processes are limited in maximum execution time in a defined period

Priority Boost Idea

NORMAL processes are boosted temporarily to RT processes, and minimum execution time is guaranteed



Patch Availability

The patch will be available in the forum patch archive.

Kprobes Implementation for Embedded System

Hitachi, Lineo Solutions

What is demonstrated

Description:Kprobes is a dynamic instrumentation system in the mainline 2.6 Linux Kernel for x86 architecture. Kprobes allows us to get information about the insight of kernel operation without rebuilding or rebooting a kernel. We will talk about the Kprobes implementation for the SH-4 architecture, and show some demos.

Operation

```
~ # insmod /lib/modules/2.6.15/kernel/drivers/char/kprobe_example.ko
```

insmod

Kprobe Handler Module

```
int init_module(void)

        int ret;
        kp.pre_handler = handler_pre;
        kp.post_handler = handler_post;
        kp.fault_handler = handler_fault;

        kp.addr = (kprobe_opcode_t*) kallsyms_lookup_name("do_fork");
        /* register the kprobe now */
        if (!kp.addr) {
                printk("Couldn't find %s to plant kprobe¥n", "do_fork");
                return -3;
        }
        if ((ret = register_kprobe(&kp) < 0)) {
                printk("register_kprobe failed, returned %d¥n", ret);
                return -2;
        }
        printk("kprobe registered¥n");
        return 0;
}
```

Replace Opcode & notifier_chain_register

Disassembled Code

```
00001dc0 <do_fork>:
   1dc0:    C3FF     mov.l    r8,@-r15
   1dc2:    53 68    mov      r5,r8
   1dc4:    96 2f    mov.l    r9,@-r15
   1dc6:    43 69    mov      r4,r9
   1dc8:    a6 2f    mov.l    r10,@-r15
   1dca:    b6 2f    mov.l    r11,@-r15
   1dcc:    63 6b    mov      r6,r11
   1dce:    c6 2f    mov.l    r12,@-r15
   1dd0:    73 6c    mov      r7,r12
   1dd2:    d6 2f    mov.l    r13,@-r15
   1dd4:    00 ed    mov      #0,r13
```

nofifier_call_chain

Kprobe Handler Function

```
int handler_pre(struct kprobe *p, struct pt_regs *regs)
{
        printk("pre_handler: p->addr=0x%p¥n",p->addr);
        dump_stack();
        return 0;
}
```

```
~ # ls
```

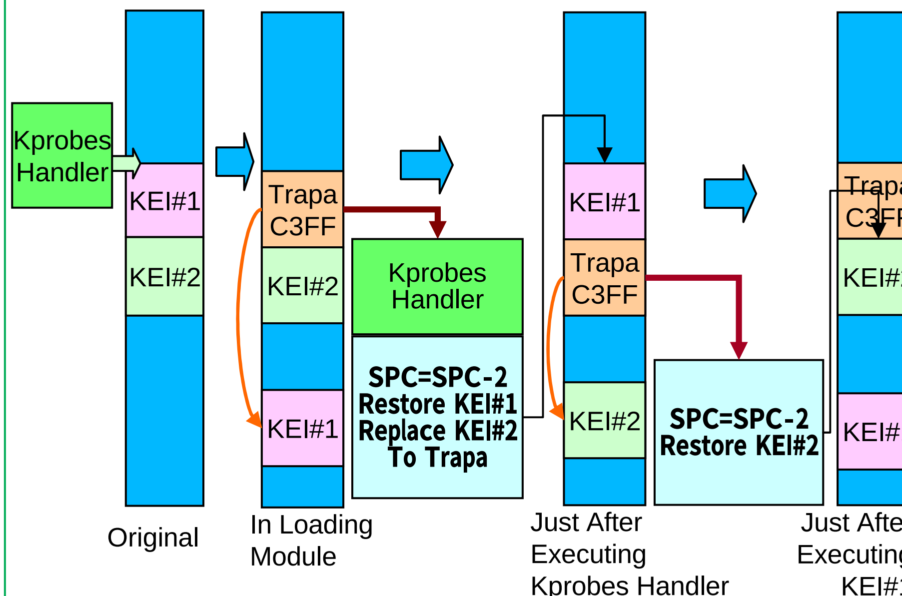
Execution Result

```
pre_handler: p->addr=0x8c021ca0

Call trace:
[<8c021ca0>] do_fork+0x0/0x240
[<8c2190a6>] kprobe_exceptions_notify+0x286/0x2c0
[<8c219162>] notifier_call_chain+0x22/0x40
[<8c012a32>] break_point_trap_software+0x32/0x80
[<8c21fa40>] __func__.2+0xb88/0x1aef0
[<8c219140>] notifier_call_chain+0x0/0x40
[<8c21fa40>] __func__.2+0xb88/0x1aef0
[<8c1713cc>] __uart_start+0x4c/0x60
[<8c0140ce>] debug_trap+0x1e/0x28
[<8c012a00>] break_point_trap_software+0x0/0x80
[<8c021ca0>] do_fork+0x0/0x240
[<8c021ca0>] do_fork+0x0/0x240
[<8c0140f4>] ret_from_exception+0x0/0xc
[<8c021ca0>] do_fork+0x0/0x240
[<8c012840>] sys_clone+0x0/0x40
[<8c021ca0>] do_fork+0x0/0x240
[<8c01285a>] sys_clone+0x1a/0x40
[<8c0141e8>] syscall_call+0xc/0xe

post_handler: p->addr=0x8c021ca0
```

Hardware Information

Renesas RTS7751R2D (SH4)

How was the Linux improved



KEI: Kernel Execution Instruction/ SPC: Save Next Program Counter

→: KEI Save / →: exception branch

Kprobes Processing in SH Implementation

Patch Availability

The patch will be available in the forum patch archive.

ARM MPCore™ and Power Management

John Goodacre

What is demonstrated

- **Includes four ARM11™ class CPUs**
  - Fully compatible with ARMv6K ISA
  - 32-bit RISC processing core with ARM and Thumb instruction sets, plus SIMD and Jazelle support
- **Synthesis configurability**
- **High efficiency coherent memory sub system**
  - Localized resolution and control over cache coherency
- **Software controlled interrupt subsystem**
  - Ability to migrate interrupt handlers to least loaded processor with integrated operating system kernel support
  - Low-latency inter-processors (IPI) and I/O interrupt distribution.
- **Low power design**
  - Minimize system overhead and power requirements
- **Compatible with existing design flows**

Hardware Information

- ARM RealView® Platform
- ARM11™ MPCore™ Processor test chip, 300Mhz

How was the Linux improved

- **Power Management**
  - Demonstrates techniques for both dynamic and static energy reduction
  - Power advantages of multiprocessing
  - Integrated into current mainline kernels
  - High performance power-aware spin locks
  - Ultra low-latency access to shared memory
- **CodeSourcery tools integration in GCC 4.1**
  - Full support for thread local storage
    - New Posix Thread Library (NPTL)
    - Memory barriers integration
  - Better than linear scalability for multitasking applications
    - Reduced context switch times
    - Improved cache utilization

Patch Availability

Patches contributed to main kernel www.kernel.org

OpenGL ES V1.1 running under ARM Linux using PowerVR MBX 3D acceleration

What is demonstrated

An ARM prototyping platform consisting of:

-ARM926EJ-S with VFP9 floating point unit
-MBX HR-S with VGP graphics acceleration block

Running a variety of example applications through the Linux version of the OpenGL ES V1.1 MBX driver.

This demonstrates the integration of OpenGL ES into an embedded Linux environment with 3D graphics acceleration.

Additionally the drivers and application make significant use of the Vector Floating Point Unit (or VFP) for geometry and physics calculations.

Hardware Information

ARM RealView® Platform for ARM926EJ-S

How was the Linux improved

- Adding OpenGL ES support provides a common API for application to access graphics functionality.
- OpenGL ES provides a fully featured graphics API in a small footprint by removing legacy functionality.
- Support for the VFP floating point unit was also added to the kernel used in this demo.

Patch Availability

ARM TrustZone™ Technology with Linux

Ian Rickards

What is demonstrated

Linux applications using secure services provided by software running in TrustZone on ARM1176JZ(F)-S.

Demonstrated with

- Linux Hack Attack demo

Other Applications Include

- Secure client-server communication (cryptography)
- Secure storage (key management and SIMLock)
- Image verification (detect OS changes/hacks)
- Secure payment services (e-commerce) and trusted User Interface.

TrustZone provides a hardware enforced separation of Normal and Secure execution worlds.

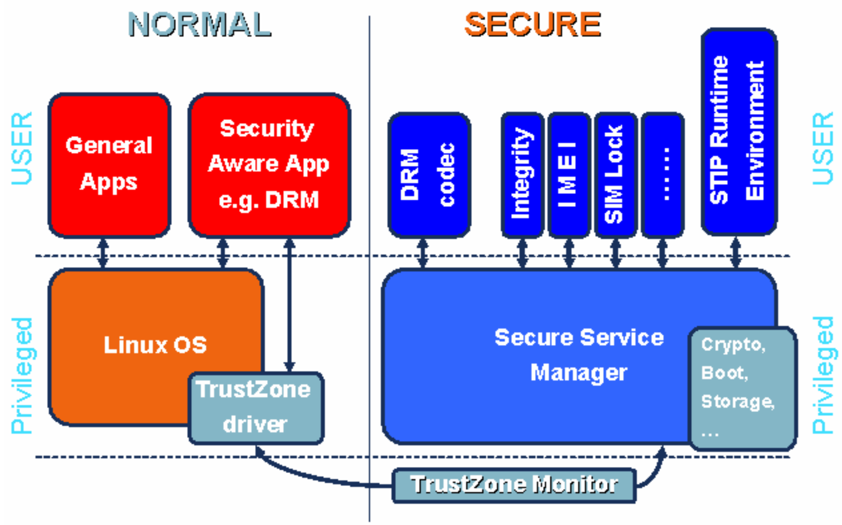
This demo uses Linux kernel 2.6.6, ARM Linux and TrustZone device driver.

Hardware Information

- RealView® Integrator™ Compact Platform
- ARM1176JZ(F)-S FPGA implementation

How was the Linux improved

- Isolated execution environment
- Secure storage (e.g. key management)
- Image verification and Secure Boot
- Crypto services to secure and transmit key data
- Secure e-commerce via secure peripherals enabling trusted user interface



Patch Availability

- ARM1176JZ(F)-S support at www.kernel.org

jazelle® - ARM Java™ Acceleration

Philippe Robin

What is demonstrated

- Direct execution of bytecode in hardware
- Highest performance Java with real applications
- Minimal memory overhead
- Low power-consumption
- Simple and quick product integration
- Robust and proven technology
- Available on a wide variety of Java technologies from a number of Java platform vendors
- Optimum solution for mobile Java platforms with Linux

Hardware Information

ARM RealView® PB-926EJ-S

How was the Linux improved

- Kernel support for Jazelle execution
  - Cache handling routines
  - Exception handling
- Fully exploit hardware capabilities available with Jazelle enabled ARM processors
- Optimized execution of Java virtual machines

NTT/DoCoMo's 902i Linux Phones using ARM11 and Jazelle

Patch Availability

www.kernel.org mainline tree

Digital Entertainment Center

ETRI

What is demonstrated

Digital Entertainment Center (DEC) is an embedded linux(Qplus) based home theater platform for sharing contents of remote PCs. DEC's several services is demonstrated.

- An extended Freevo platform for sharing contents of remote PCs on the embedded linux STB
- Remote Media Sharing Service and Remote UI Sharing Service by using UPnP technologies
- Embedded linux target configuration Method
- Browsing and streaming of AV contents by using UPnP AV architecture
- Sharing and interacting with remote PC screen by using UPnP RemoteUI architecture
- An integrated model about UPnP AV Control Point and MediaRenderer on the embedded linux
- An integrated model about UPnP RemoteUI Control Point and RemoteUI Client on the embedded linux

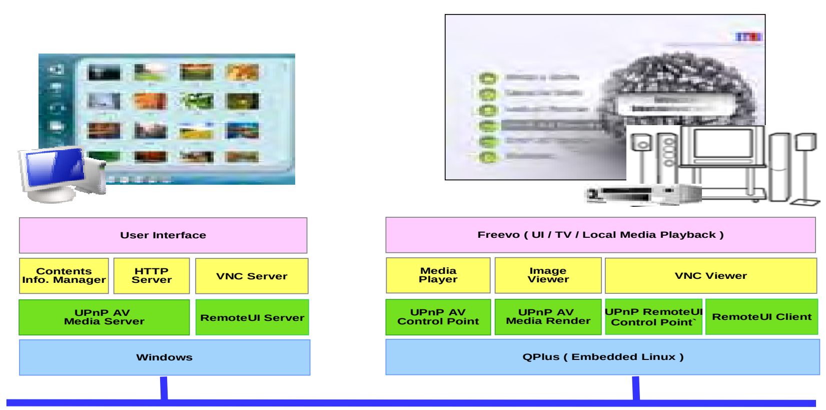
Hardware Information

- Board: Asus P5ND2
- Processor : Intel Pentium 4 Prescott 660 (3.6GHz)
- Chipset : nVIDIA nForce4
- System Memory : 1Gbyte
- VGA : Radeon X800 XL GDDR3 256MB

How was the Linux improved

We constructed DEC with Remote Media Sharing Services, Remote UI Sharing Service and UI application Programs, CELF kernel Patch, embedded linux configuration toolkit

- Various UPnP and Application technologies
- CELF released patch included
- Embedded linux target configuration toolkit (Target Builder)



Patch Availability

- Target Builder is aleady opened in CELF wiki page
- DEC related application will be available

XIP-Cramfs

Justin Treon – Intel Corporation

What is demonstrated

- Boot time comparison
- RAM usage comparison
- Performance comparison
  - Video
  - Gaming

An eXecute In Place (XIP) based system compared to a Store and Download (SnD) based system

The demonstration shows that the XIP system is actually faster that the SnD system while using less RAM

Hardware Information

XIP system: Intel M18 NOR Flash and 32MB of RAM
SnD system: NAND Flash and 64MB of RAM

How was the Linux improved

- Faster boot time
- Faster application launch time
- Reduced RAM requirement
  - Reduced bill of material

The patch allows NOR Flash based system to execute code from Flash rather than pulling applications from Flash and decompressing them in RAM

Manufacturers are able to squeeze in to the smallest RAM/FLASH combination to reduce their bill of material and boot time

Patch Availability

Publicly available, see booth handouts

Mobile Phone Based on CE Linux

NEC Corporation

What is demonstrated

N900iL
(WCDMA / Wireless LAN Dual-Mode)

•Wireless LAN Browser

N902i
(WCDMA with PoC)

•Improvement in performance of real time operations

•Reduction of application boot times

How was the Linux improved

1. Function

NEC's Linux technology has built the following mobile-phone functionality using OSS modules (SIP, RTP, RTCP, etc.). This has been achieved against a background of actively seeking external alliances with other developers.

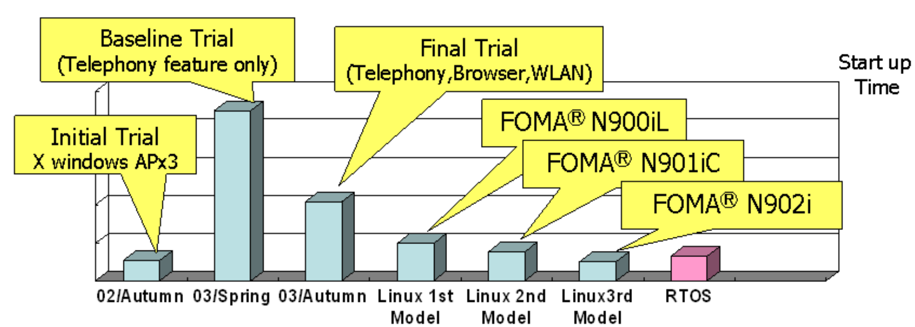
- Wireless LAN Access
- PoC (Push-To-Talk over Cellular)

Telephony API specification has been proposed to MPPWG.

2. Performance

Approaching RTOS-based phone levels of performance.

- Minimized start up time
- Reduction of application boot times



DirectFB implementation for real DTV SoC

RENESAS DirectFB (http://www.directfb.org/)

What is demonstrated

Renesas adopted DirectFB technology to the latest Digital TV SoC that support Japanese DTV standard requirement (ARIB).

DirectFB modified to support accelerated 5 layered 2D plane

DirectFB runs on Linux kernel 2.6.8-1 big-endian mode
GTK or other graphics widgets and/or various EPG browser can be combined with DirectFB to utilize 2D acceleration.

Hardware Information

SuperH processor (SH-X2) + M32R : dual core architecture incorporated with high definition support 2D graphics engine.
Note : This device is pre-production evaluation sample

How was the Linux improved

- Renesas collaborate with DirectFB project to utilize and enhance well defined opensource graphics API to support Japanese DTV

-- YUV pixel format is supported in DirectFB
-- ARIB 5 plane surface architecture support
-- Special gfx driver to utilize SoC built-in 2D acceleration and hardware scaler and bulitter

- Renesas try to combine ULDD (User Level Device Driver) with DirectFB to achieve enough stream throughput with graphics

Bench Mark Test article		A	B
Fill Rectangle	[MPixel/sec]	98.693	53.25
Fill Rectangle (blend)	[MPixel/sec]	91.776	no data
Fill Rectangles [10]	[MPixel/sec]	98.873	8.57
Fill Rectangles [10] (blend)	[MPixel/sec]	91.784	no data
Fill Spans	[MPixel/sec]	22.282	no data
Blit	[MPixel/sec]	55.593	32.56

A = Renesas DTV Soc (SH-X2 400MHz w/2D accelerator)
B = Intel Celeron 450MHz w/Matrox Millennium

Patch Availability

= Under consideration at this moment =

Direct FB would migrate to support ARIB requirement like YUV input.
Renesas plan to distribute custom gfx driver to utilize built-in 2D

Improving startup time using Software Suspend

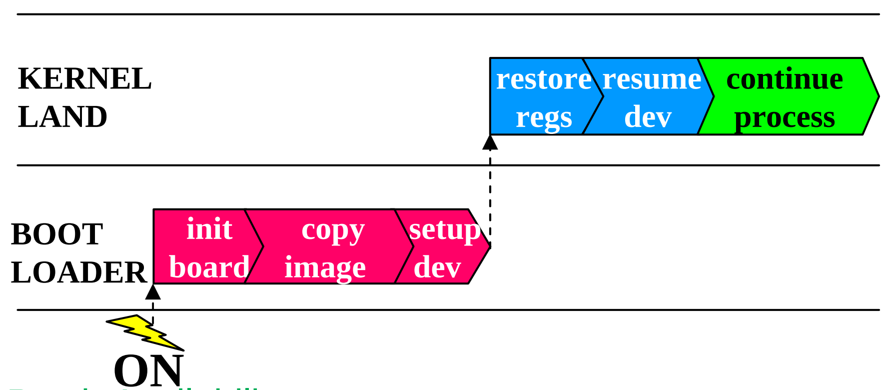
Hiroki Kaminaga

What is demonstrated

- **Startup time of each method**
  - Normal startup
    normal startup time from cold boot
  - Swsusp
    startup using software suspend, or
    suspend to flash (hibernation)
  - Snapshot boot
    startup using image created by software suspend, getting aid from boot loader to shorten startup time

How was the Linux improved

- **Startup time**
  - **swsusp support for ARM implemented**
- **Co-operation of linux and boot loader**
  - **Copying of snapshot image is done in boot loader side, and jumps to kernel-resume-point**



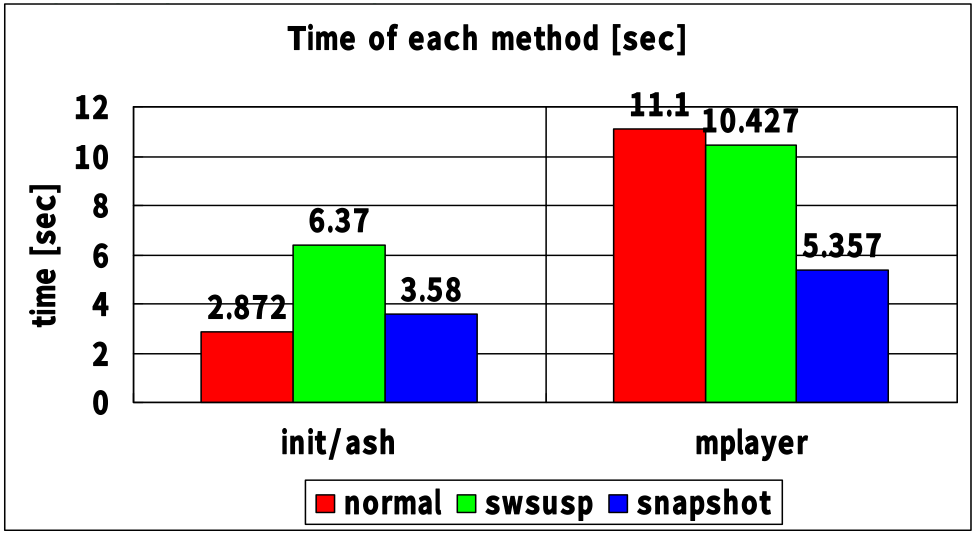
Hardware Information

- OMAP 5912 Starter Kit
  http://tree.celinuxforum.org/CelfPubWiki/OSK

Patch Availability

http://lists.osdl.org/pipermail/linux-pm/2005-July/001077

Data of each method / Issues met in snapshot boot



Measurement: CONFIG_PRINTK_TIMES

Init: time until exec() init
ash: time until thaw process
mplayer: time until thaw process,
video output for normal method.

Hardware Information

Issues met in snapshot boot

Interface between boot loader and kernel:

- many devices needed initialization and setups at boot loader side
  + lacks generalization, device power up and device resume calls in kernel should take care of this

- snapshot data structure vary at different kernel version

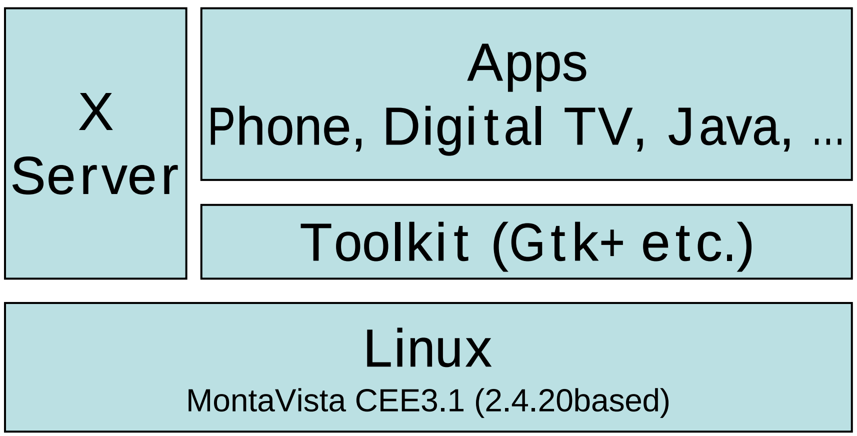
Patch Availability

Mobile Phone Powered by Linux

Masashige Mizuyama / Panasonic Mobile Communications

What is demonstrated

Mobile Phone built on top Linux and X Featuring Digital TV



Key technologies to make it work on Linux

1.Reduction of memory footprint
- -“Allocate on Write” (our original improvements)
- -XIP (eXcute In Place), ARM Thumb® code

2. Prelink to improve key response/boot time
3. Avoiding priority inversion for RT Apps by eliminating inter-threads race conditin (heap, file, mutex …)

Hardware Information

Panasonic *UniPhier*® (ARM11 core included)

How was the Linux improved

“Allocate on Write”

Defer RAM page allocation for .data until process writes to the page (Normal Linux allocates the page on ether read or write)

Implementation

No change in kernel code.
Small change to the runtime dynamic linker:

1. Drop PROT_WRITE bit when “mmap”ing ELF data segment

   By this, the kernel (CRAMFS) maps the segment to ROM page just as XIP text segment.

2. Then, set PROT_WRITE by mprotect()

   By this, copy-on-write is enable to the mapped segment memory.
   Page is copied to RAM when write occurs.
   Until then read is routed to ROM.

Patch Availability

Some patches including “Allocate on Write” and thumb® tool chain are available on CELF public Wiki pages.